AMENDMENTS TO THE SPECIFICATION

In the Office Action, the Examiner objected to the Fig. 5 because it includes reference numeral 88 not mentioned in the specification. Applicant has amended paragraph [0056] to include the proper reference to reference numeral 88. In addition, Applicant has amended paragraphs 0004, 0006, 0008, 0024, 0025, 0029, 0030, 0033, 0034, 0037, 0038, 0039, 0040, 0050 and 0056 to correct the misspelling of the term "coarse." Applicant apologizes for this error.

Please replace paragraphs 0004, 0006, 0008, 0024, 0025, 0029, 0030, 0033, 0034, 0037, 0038, 0039, 0040, 0050 and 0056 with the following amended paragraphs:

[0004] Most routers provide a form of "~~course~~coarse-grain" access control in which the internal resources are logically aggregated into groups, often in the form of a configuration hierarchy, and user access is controlled on a per-group basis. For example, one typical implementation is to assign each user an authorization level. When a user attempts to access a resource with the router, the management interface determines whether the user has a sufficient authorization level to access the resource. If the user's authorization level is sufficient, the user is permitted to access, view, or otherwise configure the resource. If not, the management interface denies access.

[0006] As the complexity of routers continually increases, there has been an increasing need to provide adequate user-level access control to the myriad of resources and information associated with a given router. For example, a typical router within the Internet, such as a router used by an Internet Service Provider, may have hundreds or even thousands of interfaces supporting thousands of different customers. These so called ~~course~~coarse-grain control techniques often do not scale well as they, for example, may require an unworkable number of authorization levels or permission bits. As a result, it is often difficult to adequately provide user-level access control to the numerous resources within a router.

[0008] Consistent with the principles of the invention, a management interface of the device supports a class syntax for defining authorization classes for controlling the access rights of remote clients. The class syntax supports a set of attributes, including a *permissions* attribute that provides ~~course~~coarse-grain access control over groups of resources. The class syntax supports two additional class attributes that may optionally be used in conjunction with the *permissions* attributes to provide fine-grain access control to the resources. In particular, an *allow-configuration* attribute and a *deny-configuration* attribute can be used to provide explicit, fine-grain access control. Each of these attributes can be associated with a respective "regular expression," which generally refers to a formula that is used to match patterns within textual data. The device applies the regular expressions associated with the fine-grain access control class attributes to evaluate text-based commands provided by the clients, and to selectively allow or deny access requests to access configuration data within the device based on the evaluation. The device may apply the regular expressions in real-time to process configuration commands as the clients enter, e.g., type, the commands.

[0024] As described in detail, network device 14 supports fine-grain access control using regular expressions. More specifically, network device 14 supports a syntax for defining authorization classes for controlling the access rights of clients 20. The class syntax supports a set of attributes, including a *permissions* attributes for providing ~~course~~coarse-grain access control. The class syntax supports two additional class attributes that may optionally be used in conjunction with the *permissions* attribute to provide fine-grain access control. In particular, an *allow-configuration* attribute and a *deny-configuration* attribute can be used to provide explicit, fine-grain access control to particular portions of the configuration data maintained by network device 14.

[0025] Each of these attributes can be associated with a respective "regular expression," which generally refers to a formula that is used to match patterns within textual data. Network device 14 applies the regular expressions associated with the fine-grain access control class attributes to evaluate text-based commands provided by clients 20, and selectively allows or denies access requests to configuration data with the network device based on the evaluation of the regular expressions and the ~~course~~coarse-grain access control attributes. In one embodiment, network device 14 applies the regular expressions in real-time to process configuration commands from clients 20 as the clients enter, e.g., type, the commands.

[0029] Management interface 38 controls access to configuration data 40 in accordance with authorization data 42, which defines authorization classes in accordance with a class syntax. Each authorization class includes a *permissions* attributes for providing ~~course~~coarse-grain access control, and optionally includes an *allow-configuration* attribute and a *deny-configuration* attribute to provide explicit, fine-grain access control to particular portions of the configuration data 40.

[0030] In one embodiment, configuration data 40 is arranged in the form of a multi-level configuration hierarchy having a plurality of inter-related objects. Each object has a textual label, e.g., a name, and represents a portion of configuration data 40 that relates to one or more resources of network device 14. Regular expressions associated with the *allow-configuration* attribute and *deny-configuration* attributes provide explicit, fine-grain access control over portions of configuration data 40 by defining textual patterns that match the textual labels of one or more of the objects within the configuration hierarchy. In this manner, the regular expressions associated with the fine-grain access control attributes may specify complex formulas identifying objects at any level of the configuration hierarchy, and are not limited to controlling access to higher-level objects, as are many conventional ~~course~~coarse-grain techniques. Control unit 32 may store configuration data 40 and authorization data 42 on one or more computer-readable media, and in the form of one or more text files, databases, tables, data structures, combinations thereof, or the like.

[0033] As illustrated, the exemplary class syntax supports a set of attributes. The *permissions* attribute represents a set of permission bits that are used for ~~course~~coarse-grain access control. In particular, each of bits is associated a high-level object within the hierarchy of configuration data 40, and provides ~~course~~coarse-grain access to the respective portions of the configuration data related to the respective objects.

[0034] In addition to the ~~course~~coarse-grain access control provided by the *permissions* attributes, the class syntax supports two class attributes that may optionally be used in conjunction with the *permissions* attribute to provide fine-grain access control to portions of configuration data 40. In particular, an *allow-configuration* attribute can be used to selectively authorize access to portions of configuration data 40 at any level of the configuration hierarchy. Similarly, a *deny-configuration* attribute can be used to selectively deny access to portions of configuration data 40 at any level of the configuration hierarchy.

[0037] The following pseudocode illustrates an example authorization class conforming to the above-described class syntax:

```
class {
      name example-user-class;
      permissions SYSTEM|FIREWALLS;
      allow-configuration "INTERFACES Ethernet fe_0/0/0";
      deny-configuration "SYSTEM login user f.*";
}
```

In this example, an authentication class is defined to have a class name of *example-user-class*. The *permissions* attribute that provides ~~course~~coarse-grain access control is set to allow users associated with the class to access portions of configuration data 40 associated with the SYSTEM object 52A object and the FIREWALLS object 52N, as well as all lower-level objects depending from these objects in the hierarchy of the configuration data 40. As a result, management interface 38 of network device 14 generally prevents users associated with this exemplary authorization class from accessing other portions of configuration data 40, e.g., the interfaces of the network device that are associated within INTERFACES object 52B and its dependent objects.

[0038] Although the users are generally not given ~~course~~coarse-grain access, the *allow-configuration* class attribute provides the users with selective authorization to access the portion of configuration data 40 corresponding to the particular Ethernet interface *fe_0/0/0* object 54. Consequently, the users are prevented from accessing any portion of configuration data 40 associated with Sonet object 56, ATM object 58, and Ethernet object 60 other than *fe_0/0/0* 54. With respect to Ethernet interface *fe_0/0/0* object 54, the *allow-configuration* class attribute authorizes the users to access portions of configuration data 40 associated with Ethernet interface *fe_0/0/0* object 54 and any lower-level objects dependent therefrom.

[0039] The *permissions* attribute grants ~~course~~coarse-grain access rights to the users for all portion of configuration data 40 related to SYSTEM object 52A, e.g., system parameters, user information, and the like. The *deny-configuration* attribute, however, instructs management interface 38 to selectively deny access to those portions of configuration data 40 that any object having a textual label that matches the regular expression "`SYSTEM login user f.*`," e.g., the portion of configuration data 40 associated with *foo* object 62. In other words, users associated with the example authorization class are allowed to access all system-related portions of configuration data 40, but specifically denied access object having a textual label that matches the regular expression "`SYSTEM login user f.*`."

[0040] FIG. 4 is a flowchart illustrating operation of network device 14 when interacting with client 20A to define an authorization class having fine-grain access control attributes. Initially, management interface 38 receives input defining an authorization class, including ~~course~~coarse-grain access control and optionally fine-grain access control attributes (70). Specifically, client 20A accesses authorization information 42 via management interface 38 and defines an authorization class, such as the exemplary authorization class *example-user-class* described above.

[0050] For the *deny-configuration* attribute, management interface 38 pre-processes the regular expression `"SYSTEM login user f.*"` to produce a more explicit regular expression `"^SYSTEM login user f.*$"`. Assuming the user has been given ~~course~~coarse-grain access to SYSTEM configuration data, management interface 38 need only apply the regular expression to pattern match for the explicitly denied commands.

[0056] If explicit authorization has been given, and the command has been submitted by client 20A, management interface 38 allows the client to proceed with the command and, when complete, processes the command (92) to display or update the accessed portion of configuration data 40 (94). If, however, explicit authorization has neither been denied or allowed via the fine-grain access control attributes, management interface 38 accesses the ~~course~~coarse -grain access control *permissions* attribute of the authorization class (88) to determine whether client 20A has been given broad authorization to access portions of configuration data 40 that encompass the specific portion being requested by the client (90). For example, in response to the following command,

```
>show system login user foo
```

management interface 38 determines whether the ~~course~~coarse -grain access control *permissions* attribute has been set to authorize client 20A to access portions of configuration data 40 associated with SYSTEM object 52A (FIG. 3) as well as all lower-level objects depending from these objects in the hierarchy of the configuration data 40. If so, management interface 38 processes the command (92). Otherwise, management interface 38 indicates the unauthorized access to client 20A, and rejects all or portions of the command (85).